

Ateliers d'initiation à L^AT_EX- Première séance

Jérôme Balthazar Pierre-Edouard Jacoby Martin Van den Abbeele
Laurent Bataille

Février 2021

Table des matières

1	Introduction	2
1.1	Qu'est-ce que L ^A T _E X	2
1.2	Pourquoi utiliser L ^A T _E X ?	2
1.3	Pourquoi ne pas utiliser L ^A T _E X	3
2	Comprendre et écrire le L^AT_EX	4
2.1	Fonctionnement	4
2.2	Interface d'Overleaf	6
2.2.1	Fichiers	6
2.2.2	Éditeur	6
2.2.3	Visionneur pdf	7
2.2.4	Menu	7
2.2.5	Journal des logs	8
2.3	Structure générale d'un document	9
3	L^AT_EX pas à pas	10
3.1	Écrire son premier document latex	11
3.2	Le préambule	13
3.3	Ajouter un titre à un document	14
3.4	Structurer un document	14
3.5	Créer une table des matières	15
3.6	Contrôler la taille du texte	15
3.7	Gérer le déplacement de blocs de textes	15
3.8	Ajouter une note en bas de page	16
3.9	Les listes et énumérations	16
3.10	Caractères spéciaux	17

1 Introduction

1.1 Qu'est-ce que $\text{\LaTeX}$

$\text{\LaTeX}$ (prononcé lay-tek or lah-tek) est un outil utilisé pour créer des documents à l'air professionnel. Il est basé sur l'idée du WYSIWYM¹, qui veut dire que vous avez seulement à vous focaliser sur le contenu de votre document et que l'ordinateur prendra en charge la mise en page selon le paramétrage du document. Au lieu d'espacer le texte d'une page pour contrôler la mise en page, comme sur Microsoft Word ou LibreOffice Writer, les utilisateurs peuvent entrer du texte brute et laisser latex se charger du reste !

1.2 Pourquoi utiliser $\text{\LaTeX}$?

$\text{\LaTeX}$ est utilisé partout dans le monde pour la mise en page de documents scientifiques, livres et aussi toutes sortes de publications. Outre la mise en page, mais il offre également aux utilisateurs des solutions pour pallier rapidement à des étapes plus chronophages de la rédaction, telles que l'insertion d'équations complexes, la réalisation d'une table des matières, le référencement de figures ou la création d'une bibliographie, tout en conservant un style homogène sur l'intégralité du document.

L'une des raisons principales pour laquelle les gens utilisent $\text{\LaTeX}$ est qu'il sépare le contenu du document de la forme. Ceci signifie qu'une fois que vous avez écrit le contenu de votre document, son apparence peut être changée facilement. De la même manière, vous pouvez créer un style de document qui pourrait vous servir de standard pour vos documents futurs : le temps passé à écrire un code versatile sera quasi-systématiquement rentabilisé. Ceci permet aux journaux scientifiques de créer des templates pour la soumission d'article par exemple². Ces templates ont une structure préétablie, dans laquelle il suffit de rajouter le texte.

Si au premier abord ces templates semblent très attirants visuellement, il est conseillé aux débutants de coder par eux-même des projets simples en $\text{\LaTeX}$, pour s'approprier à minima le langage de programmation et surtout se confronter aux erreurs les plus récurrentes survenant lors de la rédaction. La bonne nouvelle étant que la courbe de progression en $\text{\LaTeX}$ est relativement rapide et qu'aujourd'hui la communauté internet propose une aide extrêmement complète pour déboguer les inévitables messages d'erreurs. Gardez à l'esprit que la beauté du formalisme d'un langage informatique est qu'il existe rarement une seule solution à un problème !

Enfin $\text{\LaTeX}$ est assez utile pour créer des pdf interactifs, associant facilement des liens hypertextes, des vidéos ou même des vues 3D animées. À une époque où le mode de consultation préférentiel de la plupart des acteurs de la communauté scientifique est un écran, ces caractéristiques sont des atouts non-négligeables par rapport à des outils traditionnels.

1. « What you see is what you mean » . Par opposition au WYSIWYG, « What you see is what you get », correspondant aux logiciels de type Microsoft Word.

2. <http://www.latextemplates.com/template/elseviers-elsarticle-document-class>

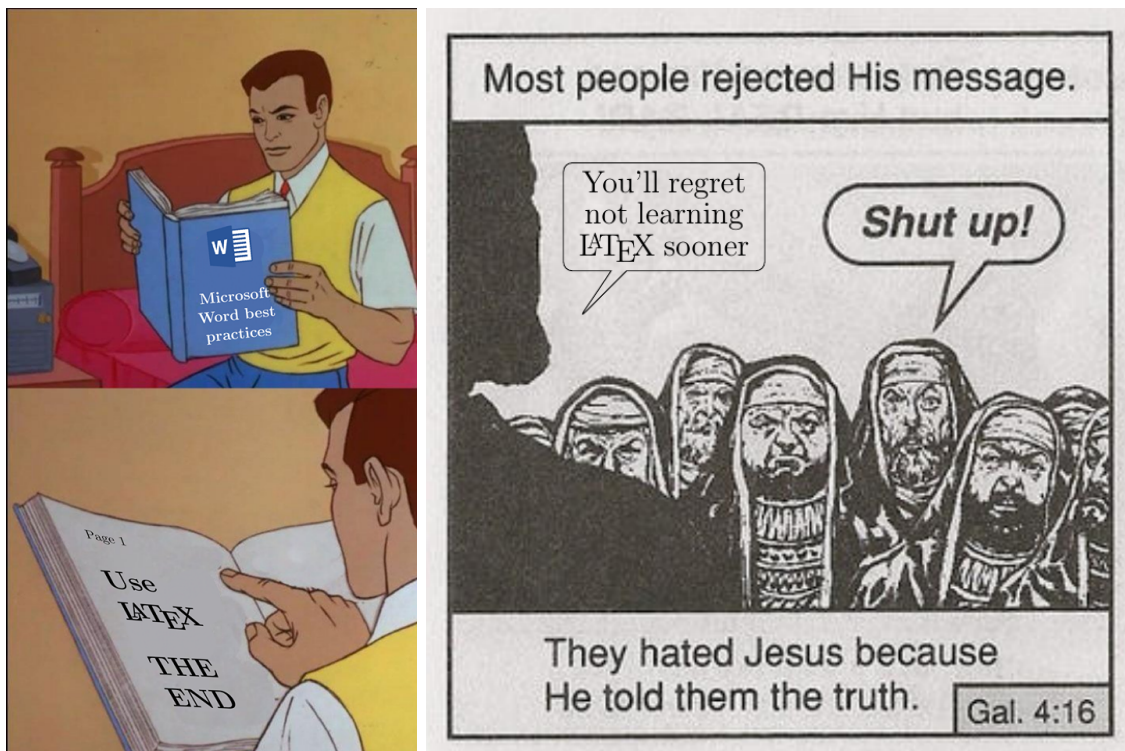


FIGURE 1 – Mêmes (pas si) honteusement copiés venant de [Mohit et al., 2019]

1.3 Pourquoi ne pas utiliser $\text{\LaTeX}$

Si le $\text{\LaTeX}$ a des avantages indéniables, il est important d’être conscient dès le départ des désavantages inhérents à ces outils/environnements.

Une étude menée par [Knauff and Nejasmic, 2014] compare la productivité horaire et le nombre de fautes commises par des débutants en environnement Word à celle d’utilisateurs au niveau de maîtrise avancé en $\text{\LaTeX}$: dans les deux cas, le résultat est significativement favorable aux utilisateurs de Word. Il est généralement préférable de commencer par écrire le contenu dans un environnement plus sobre, offrant moins sources de distraction, avant de s’attaquer directement au formatage.

Le $\text{\LaTeX}$ nécessite une phase d’apprentissage et chaque package supplémentaire ajouté par la suite, une phase d’appropriation. La rentabilité de l’investissement en temps se révèle lorsque l’utilisateur travaille sur des documents d’une certaine taille/complexité. Personnaliser un document demande davantage d’effort que d’autres interfaces, ce qui est illustré par la Figure 2.

Par défaut, le $\text{\LaTeX}$ ne compresse pas le contenu, ce qui est perçu comme un avantage pour créer des documents de qualité supérieure est perçu comme un défaut quand le document s’alourdit à l’excès et que le temps de compilation s’accroît fortement. Toutefois, en suivant quelques règles de bonne pratique et de bon sens, ces problèmes sont évitables.

Les slides réalisés sous **beamer** sont relativement sobres et linéaires. Ceci est assez subjectif, mais l’outil frustre les animateurs d’effets et de transitions travaillées en présentations. Par ailleurs, certaines manipulations requièrent par défaut un environnement WYSIWYG.

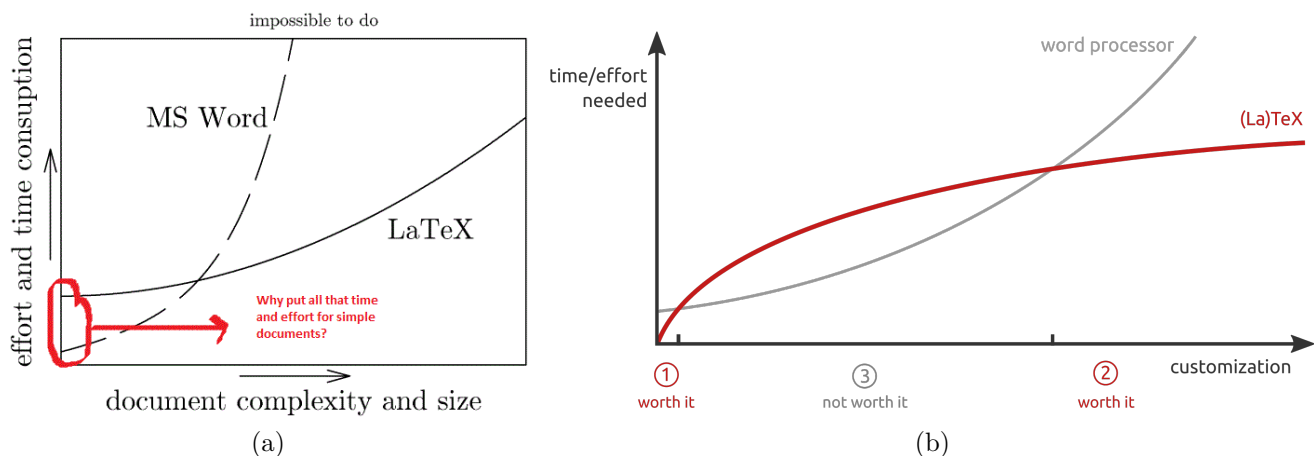


FIGURE 2 – Efforts en fonction de la complexité du document a) et du degré de personnalisation souhaité b)

Enfin, l'Enfer, c'est les autres : tout le monde n'est pas convaincu du bien fondé d'un formatage offrant un large panel de d'interaction à l'utilisateur, autant s'éviter des efforts inutiles dans certains cas.

2 Comprendre et écrire le $\text{\LaTeX}$

Avant d'attaquer la partie consacrée à la méthodologie de codage en elle-même, il est important d'aborder certains aspects, en rappelant les bases du fonctionnement général d'un langage de programmation et décrivant la mécanique interne des outils générant un document pdf à partir d'un code $\text{\LaTeX}$.

2.1 Fonctionnement

Comme tout langage de programmation le $\text{\LaTeX}$ (figure 3) fonctionne à partir d'un code (dans le cas présent `.tex`), écrit dans un éditeur donné. Un code source est un document humainement lisible/modifiable, contenant les instructions de l'utilisateur à la machine (ici pour produire un pdf). Un langage de programmation est conçu (en principe) pour être relativement ergonomique et personnalisable pour l'utilisateur, mais il n'est pas utilisable tel quel par la machine. Un logiciel réalise une traduction de ce code (ici le compilateur associé à la distribution) dans un format exploitable par la machine, c'est l'étape de compilation. La compilation est généralement combinée à une étape de vérification, où le code source est analysé par le logiciel dans le but de détecter une mauvaise utilisation de fonctions issues du langage de programmation.

Différents fichiers sont produits, certains étant des étapes intermédiaires à la construction du pdf, d'autres contiennent les informations relatives aux potentielles erreurs de compilation (`.log`). Ceci est longuement et très bien expliqué dans [Partl and Hyna, 2001] qui servira de référence lors de ces ateliers.

La formation se déroulant sur une plate-forme d'échange en ligne, ces fichiers ne sont pas directement visualisables par l'utilisateur, il est cependant préférable de connaître leur existence, particulièrement si l'on décide potentiellement de travailler hors ligne directement sur ordinateur.

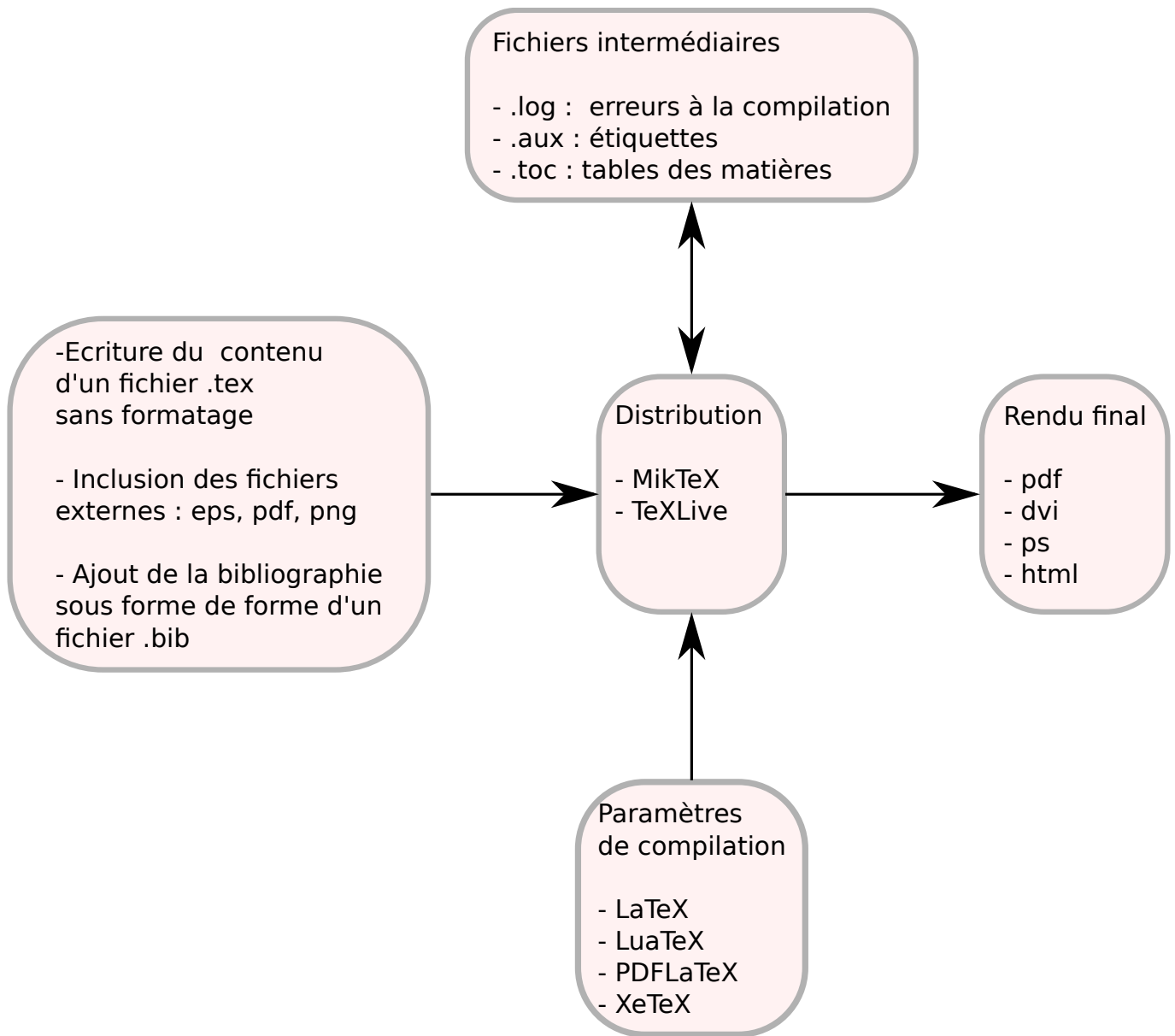


FIGURE 3 – Représentation schématique simplifiée du processus lors de la création d'un document

Par ailleurs, la plupart des environnements dédiés au $\text{\LaTeX}$ permettent de traiter ces erreurs en interne et affichent un message d'erreur généralement compréhensible par l'utilisateur.

2.2 Interface d'Overleaf

2.2.1 Fichiers

La section "Fichiers" (figure 4) à gauche permet d'importer des images de différents formats, des fichiers contenant la bibliographie en format TeX, ... Mais également de créer différents documents .tex afin de séparer différents contenus et de les importer au bon vouloir dans le document principal.

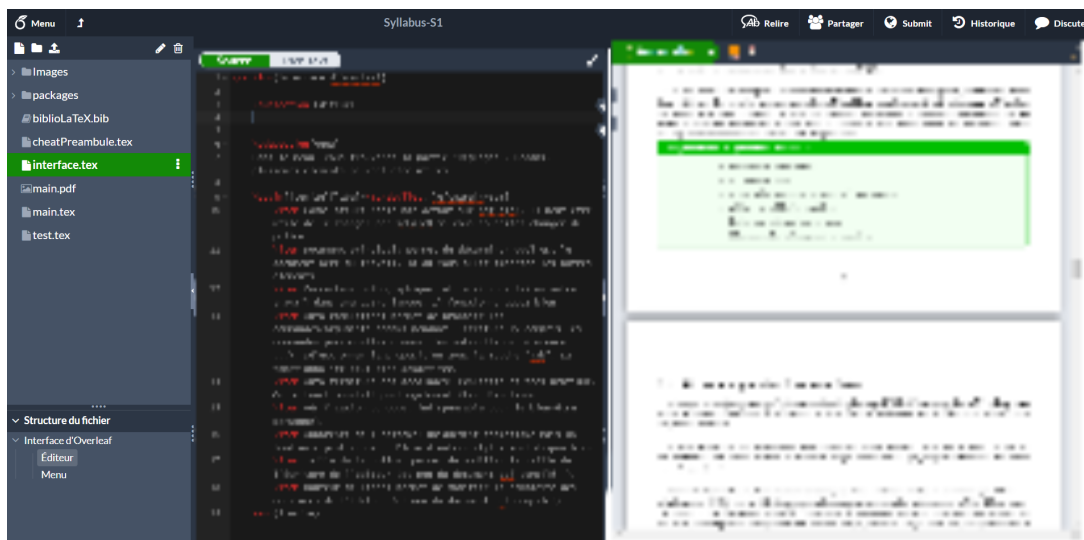


FIGURE 4 – Fichiers

2.2.2 Éditeur

L'éditeur (figure 5) est au milieu et est là où les commandes et le texte sont entrés. C'est ce code qui va être compilé pour constituer le pdf.

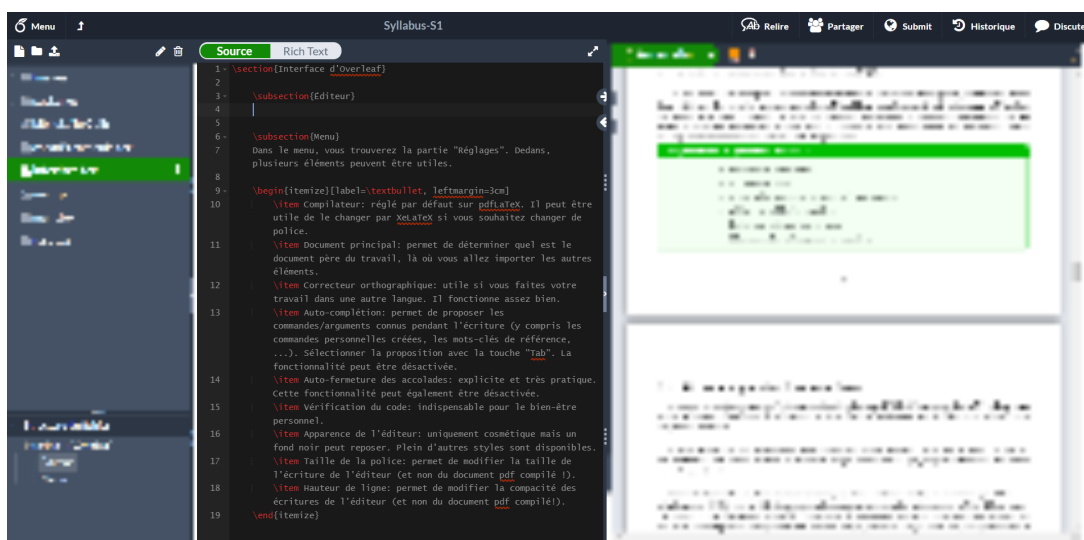


FIGURE 5 – Éditeur

Il y a la possibilité de mettre en mode riche pour la relecture. Cela met en évidence les commandes et le texte. Néanmoins, c'est plus lourd et il vaut mieux éviter de changer les commandes dans ce mode.

2.2.3 Visionneur pdf

Le visionneur (figure 6) permet de visualiser le pdf à droite suite à la compilation. La compilation "draft" compile le document plus rapidement car il ne recharge pas les figures mais juste leur position.

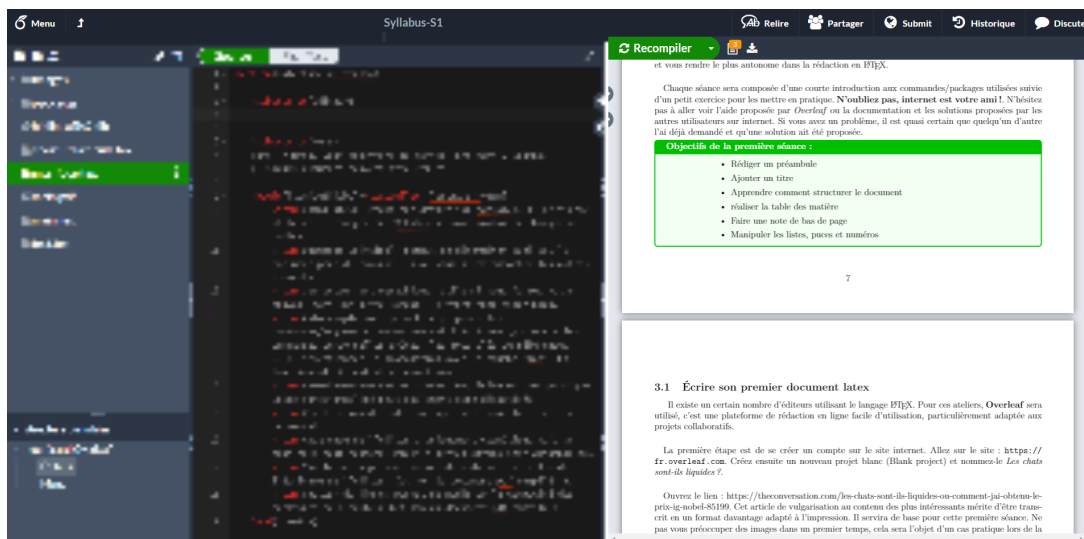


FIGURE 6 – Visionneur pdf

2.2.4 Menu

Dans le menu en haut à gauche, vous trouverez la partie "Réglages". Dedans, plusieurs éléments peuvent être utiles.

- **Compilateur** : réglé par défaut sur pdfLaTeX. Il peut être utile de le changer par XeLaTeX si vous souhaitez modifier la police.
- **Document principal** : permet de déterminer quel est le document père du travail, là où vous allez importer les autres éléments.
- **Correcteur orthographique** : utile de changer le langage de vérification si vous faites votre travail dans une autre langue. Il fonctionne assez bien.
- **Auto-complétion** : permet de proposer les commandes/arguments connus pendant l'écriture (y compris les commandes personnelles créées, les mots-clés de référence, ...). Sélectionner la proposition avec la touche "Tab". La fonctionnalité peut être désactivée si souhaité.
- **Auto-fermeture des accolades** : explicite et très pratique. Cette fonctionnalité peut également être désactivée si souhaité.
- **Vérification du code** : indispensable pour le bien-être personnel.
- **Apparence de l'éditeur** : uniquement cosmétique mais un fond noir peut reposer. Plein d'autres styles sont disponibles.
- **Taille de la police** : permet de modifier la taille de l'écriture de l'éditeur (et non du document pdf compilé!).
- **Hauteur de ligne** : permet de modifier la compacité des écritures de l'éditeur (et non du document pdf compilé!).

2.2.5 Journal des logs

Il s'agit de la fenêtre vous indiquant les éventuelles erreurs apparaissant à la compilation pour vous aider à déboguer votre code. Elle est accessible en cliquant sur l'icône entourée en rouge sur la Figure 7.

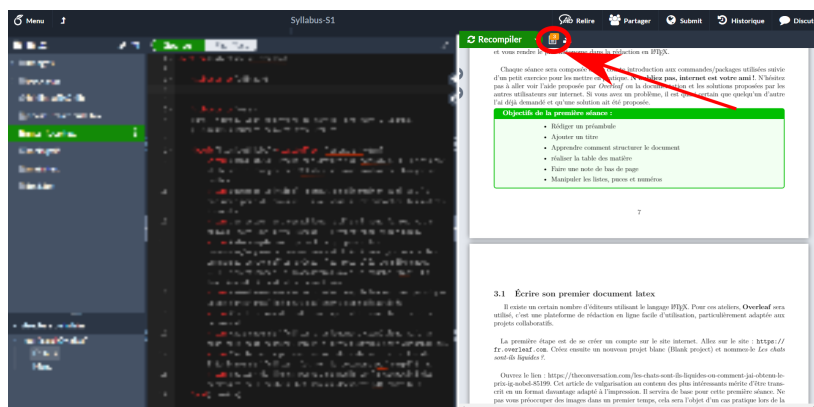


FIGURE 7 – Bouton donnant accès au journal des logs

Il existe deux catégories de messages, les avertissements (en orange), où le compilateur vous notifie qu'une manoeuvre pourrait être améliorée bien qu'un document soit compilé et les erreurs (en rouge) indiquant quels morceaux de code font planter le document, dont deux exemples sont données à la Figure 8.

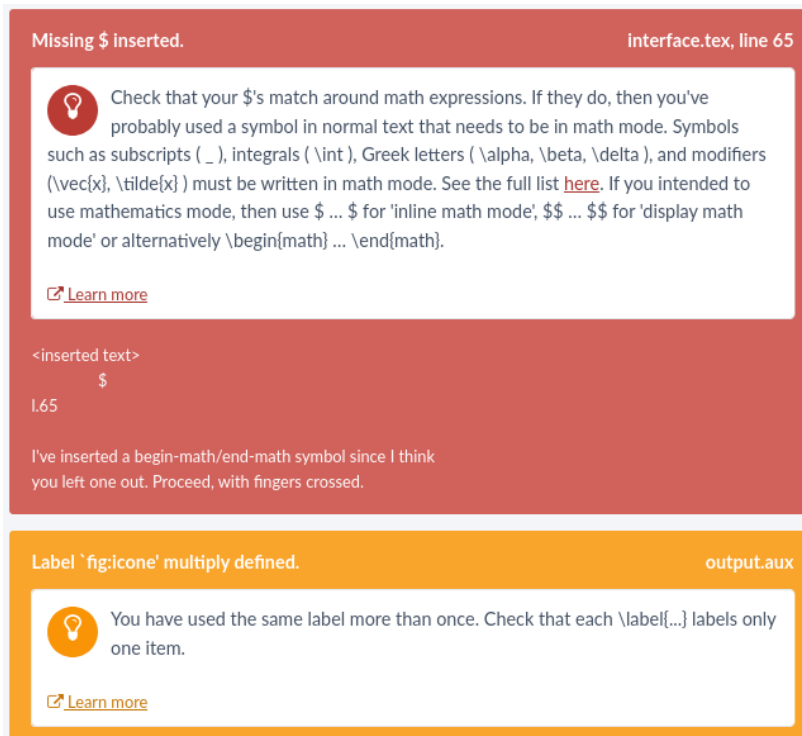


FIGURE 8 – Contenu du journal des logs

2.3 Structure générale d'un document

Tout document $\text{\LaTeX}$ est segmenté en différentes parties. Le langage s'est très fortement étoffé, au fil des années, la communauté scientifique a progressivement créé une pléiade de boîtes à outils, appelées extensions³, pour réaliser des documents très spécifiques, que ce soit par souci de personnalisation ou au contraire de standardisation vis à vis des bonnes pratiques prônées par la communauté scientifique.

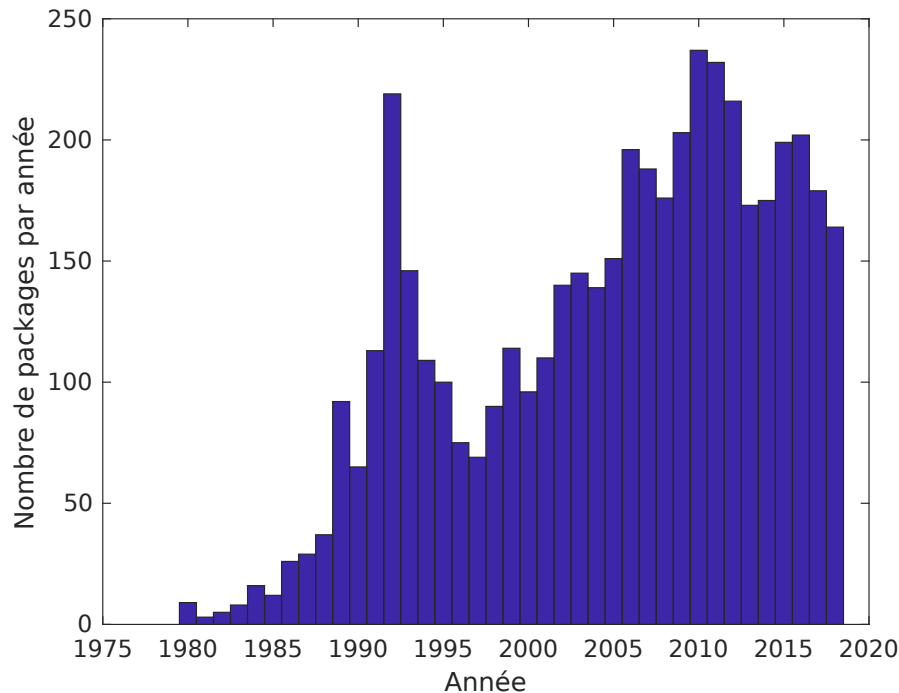


FIGURE 9 – Nombre d'extensions mises en ligne annuellement

Sur PC, $\text{\LaTeX}$ est toujours associé à un gestionnaire de packages associé à une distribution⁴, qui télécharge automatiquement les packages non-installés à partir de la base de données de référence.

3. Ces extensions et leurs notices explicatives, sont accessibles gratuitement en ligne sur une base de données, The Comprehensive The Comprehensive TEX Archive Network, ce qui est très utile lors de l'utilisation d'un nouvel outil. 6000 extensions sont disponibles et ce n'est pas fini, voir Figure 9 !

4. MiK $\text{\TeX}$ pour Windows et Mac, T $\text{\E}$ XLive pour Linux. T $\text{\E}$ XLive est par ailleurs l'outil utilisé par Overleaf

Structure minimale d'un .tex

```

\documentclass[options_doc]{type_de_document}
\usepackage[options1a,options1b,options1c]{package1}
\usepackage[options2a,options2b]{package2}
...
\usepackage[optionsNa,...,optionsNn]{packageN}
\begin{document}
\begin{environnement1}[optionsE1]{argE1}
    \commande1{arg1}{arg2}
\end{environnement1}
\end{document}

```

L'exemple fourni ci-dessus n'est pas exécutable et a pour but de vous indiquer la structure des commandes/environnements et de clarifier quelques points de vocabulaire !

La partie du document où sont spécifiés les packages utilisés est appelée préambule. Un préambule est constitué par toutes les lignes de codes qui sont écrites au-dessus de `\begin{document}`, tandis que le corps du document est constitué par tout ce qui est compris entre `\begin{document}` et `\end{document}`. La première ligne d'un préambule sert toujours à spécifier quel genre de document vous souhaitez produire mais nous y reviendrons plus loin.

Tout ce qui débute par un `\` est appelé une commande et aura une influence sur la compilation du document ^a, les structures introduites par `\begin{...}` et `\end{...}` sont appelées environnements, il s'agit de blocs de code obéissant à des règles de mise en forme particulières. Qu'il s'agisse de commandes ou d'environnement, les deux peuvent recevoir des options ^b et éventuellement des arguments ^c. Les options sont introduites entre crochets [...] et les arguments entre accolades {...}.

a. Pour introduire un caractère `\` dans du texte sans influencer sur la compilation, il faut utiliser la commande `textbackslash`

b. Ce qui est optionnel est... facultatif...

c. Obligatoires!!!

Charger à chaque compilation l'ensemble des packages serait très long et contre-productif pour l'utilisateur. La première étape est donc de définir le type de document ainsi que les packages spécifiques que l'on souhaite potentiellement utiliser pour produire ce document.

Au fil du temps, vos préambules s'étofferont mais gardez à l'esprit qu'en cas d'erreur à déboguer liée à un package spécifique, un préambule simple est préférable à un préambule surchargé.

3 L^AT_EX pas à pas

Cette formation sera composée de 4 séances, à travers lesquelles nous allons tenter de survoler les bases de la rédaction et de l'utilisation des commandes/packages les plus utilisés pour la rédaction de rapports scientifiques. Le but est de vous offrir les bases nécessaires pour vous lancer dans ce langage et vous rendre le plus autonome dans la rédaction en L^AT_EX.

Chaque séance sera composée d'une courte introduction aux commandes/packages utilisées suivie d'un petit exercice pour les mettre en pratique. **N'oubliez pas, internet est votre ami !**. N'hésitez pas à aller voir l'aide proposée par *Overleaf* ou la documentation et les solutions proposées par les autres utilisateurs sur internet. Si vous avez un problème, il est quasi certain que quelqu'un d'autre l'ai déjà demandé et qu'une solution ait été proposée.

Objectifs de la première séance :

- Rédiger un préambule
- Ajouter un titre
- Apprendre comment structurer le document
- réaliser la table des matières
- Faire une note de bas de page
- Manipuler les listes, puces et numéros

3.1 Écrire son premier document latex

Il existe un certain nombre d'éditeurs utilisant le langage L^AT_EX. Pour ces ateliers, **Overleaf** sera utilisé, c'est une plateforme de rédaction en ligne facile d'utilisation, particulièrement adaptée aux projets collaboratifs.

La première étape est de se créer un compte sur le site internet. Allez sur le site : <https://fr.overleaf.com>. Créez ensuite un nouveau projet blanc (Blank project) et nommez-le *Les chats sont-ils liquides ?*.

Ouvrez le lien : <https://theconversation.com/les-chats-sont-ils-liquides-ou-comment-jai-obtenu-le-prix-ig-nobel-85199>. Cet article de vulgarisation au contenu des plus intéressants mérite d'être transcrit en un format davantage adapté à l'impression. Il servira de base pour cette première séance. Ne pas vous préoccuper des images dans un premier temps, cela sera l'objet d'un cas pratique lors de la prochaine séance !

Pour utiliser efficacement Overleaf, n'hésitez pas à vous référer au guide des raccourcis claviers à la figure 10 et disponible à l'adresse suivante : <https://www.overleaf.com/learn/how-to/Hotkeys>.

Command	Operation
Ctrl + f	Find and replace.
Ctrl + Enter	Compile the project.
Ctrl + z	Undo.
Ctrl + y	Redo.
Ctrl + Home	Go to first line of the current file.
Ctrl + End	Go to last line of the current file.
Ctrl + l	Go to specific line (opens a dialog)
Ctrl + /	Toggle commenting for selected lines. (Comment/uncomment selected text.)
Ctrl + d	Delete current line.
Ctrl + a	Select all.
Tab	Indent section.
Shift + Tab	Un-indent section.
Ctrl + u	Change selected text to upper case.
Ctrl + Shift + u	Change selected text to lower case.
Ctrl + b	Bold.
Ctrl + i	Italics.
Ctrl + Space	Open autocomplete menu, inside of <code>\cite{...}</code> search the bibliography.

FIGURE 10 – Raccourcis clavier sur Overleaf

3.2 Le préambule

Rédigez le préambule de ce document en sachant que le document est un article, rédigé sur une feuille au format A4, avec une police de base de 12 pt.

Premier document - N'est-ce pas mignon ? 

```
\documentclass[a4paper,12pt]{article} % Options de style et type de document...
    Toujours en première ligne de votre fichier main.
\usepackage[utf8]{inputenc}           % Encodage utilisé pour le fichier source
\usepackage{ae,lmodern}               % Polices vectorielles et police adaptée au
    français
\usepackage[french]{babel}            % Langue(s) utilisées dans le document
\usepackage[T1]{fontenc}              % Caractères utilisés

\usepackage[linktocpage]{hyperref}    % Créations de liens hypertextes
% On écrit les commentaires avec le symbole pourcent en LaTeX
% Pour commenter plusieurs lignes, préalablement sélectionnées
% le raccourci clavier est ctrl + / sur Overleaf

% Taille des marges
\usepackage{geometry}
\geometry{
    includeheadfoot,
    top=1.5cm,
    bottom=1.5cm,
    right=2.00cm,
    left=2.00cm}
\begin{document}
    blabla
%Insérer contenu... Sans texte, tout plante car générer un pdf vide n'a pas de sens
\end{document}
```

La première ligne permet de décrire le type de document⁵, ce qui aura surtout une influence sur les commandes disponibles pour scinder le document en plusieurs parties et sous-parties, ainsi que les options de styles qui y sont associées. Les options de style par défaut sont reprises dans la Table 1

5. Par exemple les types `book`, `report`, `article`, mais il en existe beaucoup d'autres !

TABLE 1 – Synthèse des options de style disponibles

Objet	Options (par défaut)
Taille du papier	a4paper, a5paper, b5paper, <u>letterpaper</u> , legalpaper, executivepaper
Taille des lettres	<u>10 pt</u> , 11 pt, 12 pt
Orientation	landscape, <u>portrait</u>
Page de titre séparée	<u>titlepage</u> , notitlepage
Version du document	draft, <u>final</u>
Simple/double face	<u>oneside</u> , twoside
Position du début des chapitres	openright, openany
Colonnes (une ou deux)	<u>onecolumn</u> , twocolumn

3.3 Ajouter un titre à un document

Pour ajouter un titre à un document, il existe plusieurs méthodes. La plus simple et proposée par Overleaf est de définir le titre, le(s) auteur(s) et la date dans le préambule et d'utiliser la commande `\maketitle`. Cette commande s'insère bien après le `\begin{document}`.

Ajout d'un titre - exemple simple

```
....
\title{Les chats sont-ils liquides ou comment j'ai obtenu le prix Ig Nobel}
\author{Marc-Antoine Fardin}
\date{Novembre 2017}

\begin{document}
\maketitle
...
```

Une fois cette partie rédigée, recompiler pour voir le résultat. Il est possible de changer la taille/-format du titre à l'aide d'un panel de commande. Il est aussi possible de définir des pages de garde un peu plus étoffées avec l'environnement `titlepage`.

3.4 Structurer un document

L^AT_EX est un très bon environnement pour structurer son texte et hiérarchiser les contenus. Utiliser les commandes suivantes pour structurer l'article de référence :

Structure d'un document book

```

\chapter{Mon titre de chapitre} % Chapter ne fonctionne pas dans l'environnement
    article, on ajoute des chapitres dans des livres, mais pas des articles !
\section{Mon titre de section}
% * = Pas de numérotation et n'apparaît pas dans la table des matières !
\section*{Mon titre de section non-numérotée}
% [] : arguments optionnels, ici on peut définir un titre plus court
% pour la table des matières et l'arborescence finale du pdf
\subsection[Plus petit titre]{Mon titre de sous-section}
\subsubsection{Mon titre de sous-sous-section}
\paragraph{} % Non numéroté mais c'est plus propre pour séparer des paragraphes

```

3.5 Créer une table des matières

Les tables des matières peuvent être une étape fastidieuse de la finalisation d'un rapport dans le cas de gros projets, potentiellement soumis à modifications de dernière minute. Cependant une commande simple permet d'en établir une de façon simple et rapide : utiliser `\tableofcontents`.

3.6 Contrôler la taille du texte

La taille du texte est spécifiée par une commande, par exemple `\tiny{Minuscule}`. L'ensemble des tailles sont reprises à la Table 2.

TABLE 2 – Contrôler la taille du texte - commandes

Commande	Exemple	Taille en 10 pt
<code>\tiny</code>	Minuscule	5
<code>\scriptsize</code>	Très, très petite	7
<code>\footnotesize</code>	Très petite	8
<code>\small</code>	Petite	9
<code>\normalsize</code>	Normale, valeur par défaut	10
<code>\large</code>	Plus grande que la normale	12
<code>\Large</code>	Grande	14.4
<code>\LARGE</code>	Très grande	17.28
<code>\huge</code>	Enorme	20.74
<code>\Huge</code>	Gigantesque	24.88

3.7 Gérer le déplacement de blocs de textes

- Pour indiquer un retour à la ligne il existe plusieurs commande `\newline` ou la version courte `\\`.
- Un paragraphe se commence souvent en indentant le texte, ce qui se réalise via la commande `\paragraph{}`. Pour supprimer une indentation, la commande `\noindent` a fait ses preuves.
- Enfin pour imposer un saut de page, on utilise `\newpage`.

3.8 Ajouter une note en bas de page

L^AT_EX permet d'ajouter très simplement des notes en bas de page via la commande :

```
\footnote{Ma sublimissime note de bas de page}.
```

3.9 Les listes et énumérations

L'exemple le plus basique est la création d'une liste via l'environnement `itemize` et de la commande `\item`. L'extension `enumitem` facilite grandement la manipulation des listes, il est nécessaire de la charger via la commande `\usepackage{enumitem}`. Ce package permet par exemple de

Liste - contrôle des puces par l'environnement `enumitem`

```
Exemple de liste :
\begin{itemize}
\item[$\cdot$] Element 1
\item[$\cdot$] Element 2
\item[$\cdot$] Element 3
\end{itemize}
% Option label via enumitem pour l'ergonomie
Liste non-exhaustive des siphonaptères
\begin{itemize}[label=$\cdot$] % On spécifie le type de puce de la liste via l'
    option label de itemize
\item Puce du rat
\item Puce-chique
\item[$\star$] Puce de l'homme % On peut le faire aussi à la main dans les option
    de \item
\end{itemize}
```

L'énumération se réalise avec l'environnement `enumerate`, qui reste par ailleurs très proche de `itemize`. Il est possible de sélectionner le type de numérotation et l'indentation dans les options de `enumerate`.

Exemple d'utilisation de `enumerate`

```
Quelques masters de Gembloux classés aléatoirement
\begin{enumerate}[label=\arabic*]
\item Sciences et technologies de l'environnement
\begin{enumerate}[label=\alph* --,leftmargin=1.5cm]
\item Génie rural
\item Gestion du territoire
\end{enumerate}
\end{enumerate}
\item Gestion des forêts et des espaces naturels
\item Chimie et bioindustries
\item Sciences agronomiques
\end{enumerate}
```

Nous vous invitons à retirer les options des énumérations imbriquées les unes dans les autres pour découvrir le fonctionnement par défaut de l'environnement `enumerate`.

3.10 Caractères spéciaux

Voici les symboles possibles à utiliser sur overleaf (figure 11).

$\leq$	<code>\leq</code>	$\propto$	<code>\propto</code>	$\$$	<code>\\$</code>	α	<code>\alpha</code>	∞	<code>\infty</code>
$\geq$	<code>\geq</code>	$\cdots$	<code>\cdots</code>	$\&$	<code>\&</code>	β	<code>\beta</code>	f'	<code>f'</code>
$\neq$	<code>\neq</code>	$\dots$	<code>\dots</code>	$\%$	<code>\%</code>	ϵ	<code>\epsilon</code>	$\int$	<code>\int</code>
$\nleq$	<code>\nleq</code>	$\because$	<code>\because</code>	$\backslash$	<code>\backslash</code>	ζ	<code>\zeta</code>	$\oint$	<code>\oint</code>
$\ngeq$	<code>\ngeq</code>	$\therefore$	<code>\therefore</code>	$\sharp$	<code>\sharp</code>	η	<code>\eta</code>	$\mathbb{Z}$	<code>\mathbb{Z}</code>
$\cong$	<code>\cong</code>	$\forall$	<code>\forall</code>	∂	<code>\partial</code>	κ	<code>\kappa</code>	$\mathbb{R}$	<code>\mathbb{R}</code>
$\equiv$	<code>\equiv</code>	$\exists$	<code>\exists</code>	90°	<code>90^\circ</code>	λ	<code>\lambda</code>	$\mathbb{Q}$	<code>\mathbb{Q}</code>
$\sim$	<code>\sim</code>	$\in$	<code>\in</code>	$\parallel$	<code>\parallel</code>	μ	<code>\mu</code>	$\sqrt[3]{2}$	<code>\sqrt[3]{2}</code>
$\approx$	<code>\approx</code>	$\subset$	<code>\subset</code>	$\perp$	<code>\perp</code>	ξ	<code>\xi</code>	$\frac{2}{3}$	<code>\frac{2}{3}</code>
$\doteqdot$	<code>\doteqdot</code>	$\subseteq$	<code>\subseteq</code>	$\triangle$	<code>\triangle</code>	ρ	<code>\rho</code>	$\lceil x \rceil$	<code>\lceil x \rceil</code>
$\times$	<code>\times</code>	$\varnothing$	<code>\varnothing</code>	∇	<code>\nabla</code>	τ	<code>\tau</code>	$\lfloor x \rfloor$	<code>\lfloor x \rfloor</code>
$\cdot$	<code>\cdot</code>	$\cap$	<code>\cap</code>	$\square$	<code>\square</code>	ϕ	<code>\phi</code>	$\{x\}$	<code>\{x\}</code>
$\ast$	<code>\ast</code>	$\cup$	<code>\cup</code>	$\angle$	<code>\angle</code>	ψ	<code>\psi</code>	$\widehat{p}$	<code>\widehat{p}</code>
$\div$	<code>\div</code>	$\setminus$	<code>\setminus</code>	Π	<code>\Pi</code>	π	<code>\pi</code>	$\overline{AB}$	<code>\overline{AB}</code>
$\pm$	<code>\pm</code>	$\wedge$	<code>\wedge</code>	Θ	<code>\Theta</code>	θ	<code>\theta</code>	$\overleftrightarrow{AB}$	<code>\overleftrightarrow{AB}</code>
$\mp$	<code>\mp</code>	$\vee$	<code>\vee</code>	Γ	<code>\Gamma</code>	γ	<code>\gamma</code>	$\overset{\small{AB}}{\frown}$	<code>\overset{\small{AB}}{\frown}</code>
$\bigcirc$	<code>\bigcirc</code>	$\Rightarrow$	<code>\Rightarrow</code>	Δ	<code>\Delta</code>	δ	<code>\delta</code>	$\overline{\overline{AB}}$	<code>\overline{\overline{AB}}</code>
$\oplus$	<code>\oplus</code>	$\rightarrow$	<code>\rightarrow</code>	Ω	<code>\Omega</code>	ω	<code>\omega</code>	$\overline{\overline{\overline{AB}}}$	<code>\overline{\overline{\overline{AB}}}</code>
$\otimes$	<code>\otimes</code>	$\mapsto$	<code>\mapsto</code>	Σ	<code>\Sigma</code>	σ	<code>\sigma</code>	$\overline{\overline{\overline{\overline{AB}}}}$	<code>\overline{\overline{\overline{\overline{AB}}}}</code>

FIGURE 11 – Symboles

Par contre, certains caractères sont utilisés dans le langage $\text{\LaTeX}$ et donc ne peuvent pas servir directement de contenu, sous peine d'un message d'erreur. Pour pallier à cela, plusieurs options sont disponibles. Il suffit la plupart du temps de placer un backslash avant le terme

Caractères spéciaux

La phrase : " Kleenex© a perdu 34 % de vente depuis la fin de GOT™ " s'écrit :
`Kleenex\copyright a perdu 34 \% de vente depuis la fin de GOT\texttrademark`

Vous pouvez trouver des listes de symboles facilement sur internet, mais pour vous faciliter la tâche, voici quelques liens :

- <https://openclassrooms.com>
- <https://latex.developpez.com/faq/?page=Les-accent-et-les-caracteres-speciaux>

Références

- [Knauff and Nejasmic, 2014] Knauff, M. and Nejasmic, J. (2014). An efficiency comparison of document preparation systems used in academic research and development. *PLoS ONE*, 9(12) :1–12.
- [Mohit et al., 2019] Mohit, A., Picard, L., and Abarbanel, D. (2019). $\text{\LaTeX}$ memes for well typeset teens.
- [Partl and Hyna, 2001] Partl, H. and Hyna, I. (2001). *Une courte (?) introduction à $\text{\LaTeX}2\epsilon$* . LAAS/CNRS.

Suppléments

Installer L^AT_EX sur son ordinateur pour travailler hors ligne.

1. En fonction de votre OS, installer une distribution compatible :
 - Windows : MikTeX (<https://miktex.org/download>)
 - Linux : TeXLive (installation depuis le dépôt)
 - Mac OS X : MacTeX (<http://www.tug.org/mactex/>)
2. Si un éditeur de texte basique comme Notepad permet d'ores et déjà de coder en L^AT_EX, il est recommandé d'utiliser un éditeur spécialisé au choix, toujours selon la compatibilité de votre OS :
 - TeXMaker (<https://www.xmlmath.net/texmaker/>)
 - TeXnicCenter (<http://www.texniccenter.org>)
 - Kile (<https://kile.sourceforge.io>)

Attention à bien lire les infos pour paramétrer correctement l'éditeur lors du premier lancement.

Souligner le titre des sections facilement

Pour souligner du texte, il est possible d'utiliser la commande `\underline{texte à souligner}`. Cependant cette commande ne peut être utilisée pour les titres de section : basiquement les commandes sont utilisées pour effectuer une opération sur un bloc de texte défini par l'utilisateur dans le corps du document. Or par souci de cohérence, la commande `section` ne permet pas de personnaliser la mise en forme directement dans le corps du document, il faut définir une fois, pour l'intégralité du document, les règles du jeu dans le préambule.

Pour réaliser cette action, il est nécessaire de faire appel au package `titlesec` : il s'agit d'un outil permettant de générer une mise en forme particulière pour les titres dans un document.

Préambule - personnalisation des titres de section

```
\usepackage{titlesec}
% \titleformat{niveau de text}{Taille et forme du texte}{Appel à la variable
    comptant les éléments d'un même niveau}{Espace entre les lignes}{}[{\Commande
    inclue dans titlesec tirant une ligne de 0.8pt d'épaisseur}]
\titleformat{\section}
{\normalfont\Large\bfseries}{\thesection}{1em}{}[{\titlerule[0.8pt]}]
```

Source : <https://tex.stackexchange.com/a/314010>

Modifier la police d'écriture

La plupart du temps les utilisateurs de L^AT_EX se contentent de la police par défaut, il est toutefois naturel d'avoir envie de pousser le degré de personnalisation de son document un cran plus loin. Pour

introduire une police particulière, il suffit d'inclure le package avec le nom de cette police dans le préambule du document.

Modification de la police d'un document

```
\usepackage{tgbonum}
\begin{document}
```

L'ensemble des polices utilisables en L^AT_EX est disponible via ce lien :
https://fr.overleaf.com/learn/latex/Font_typefaces#Reference_guide

Créer un environnement qui indente le texte et le met en italique

De la même manière que dans le premier cas, une manière élégante est de paramétrer un style particulier dans le préambule pour l'utiliser de façon répétée. Dans le cas où on veut créer un moyen de mettre en page un bloc de texte particulier, la méthode la plus adaptée est de créer un nouvel environnement. Pour faire appel à un environnement, toujours dans le corps du texte, on écrit ceci de la manière suivante

Exemple de commande faisant appel à un environnement

```
\begin{nom_de_l_environnement}
texte sur lequel est appliqué la mise en forme
\end{nom_de_l_environnement}
```

Pour définir un environnement, on utilise la commande `\newenvironment`

Syntaxe d'un environnement

```
% Le nombre d'argument est optionnel
\newenvironment{nom_de_l_environnement}[nombre_arguments]{action_au_debut_de_
lenvironnement}{action_a_la_fin_de_lenvironnement}
```

Pour illustrer ce concept, reprenons le cas pratique posé en utilisant et en encapsulant différents environnements :

- minipage : pour définir un bloc de texte d'une largeur inférieure à celle de la page.
- flushright et flushleft : pour modifier l'alignement d'un bloc de texte.
- slshape : pour mettre en italique un bloc de texte

Le nom de notre environnement est `abstract_sect` et pour le définir, on peut ajouter le code suivant dans le préambule.

Encapsulation progressive des environnements -

```
% Encapsulation progressive des environnements
% Le nombre d'argument est optionnel, d'où les [], vu qu'aucun argument n'est spé
% cifié on omet d'en spécifier le nombre
\newenvironment{abstract_sect}{
\begin{flushright}    % Pour indenter on décale les blocs de texte vers la droite
\begin{minipage}{0.85\textwidth} % La commande \textwidth permet d'obtenir une
    valeur numérique décrivant la largeur de la page
\begin{flushleft} % On revient à un alignement classique dans le bloc de texte
\begin{slshape}    % Environment mettant tout son contenu en italique
}{
% On ferme tous les environnements à la fin
\end{slshape}
\end{flushleft}
\end{minipage}
\end{flushright}}
```

Pour faire un test du résultat, il faut l'appeler dans le corps du document.

Utilisation de `abstract_sect`

```
\begin{abstract_sect}
    Mettre des environnements dans les environnements, cela paraît lourd quand on
    débute, mais à l'échelle d'un gros document, c'est vite rentabilisé.
\end{abstract_sect}
```

Source : <http://www.tug.dk/FontCatalogue/latinmodernmono/>

Créer une page de titre sans utiliser la commande `maketitle`

Concrètement cela revient à mettre en page soit même. Classiquement le type de code que l'on écrit une fois dans sa vie en y mettant du cœur puis qu'on copie-colle joyeusement dans ses rapports... Il existe un environnement pour préciser au logiciel ce que l'on fait, de façon par exemple à ne pas inclure cette page dans la table des matières. Cet environnement se nomme `titlepage`.

Page de titre

```
% \begin{document}
\begin{titlepage}
    Sublime contenu de la page de titre
\end{titlepage}
```

Les commandes de taille citées plus haut sont indispensables, mais d'autres commandes et environnements pourraient vous être utiles :

- Centrer du texte ou repousser du texte en bas de page :

Mise en forme de la page de titre - exemple

```
% Pour centrer horizontalement, il existe l'environnement \verb!center!
\begin{center}
  \Huge{Mon texte gigantesque à centrer}
\end{center}
% Pour centrer verticalement, on comble l'espace vide au-dessus et en-
  dessous avec la commande \vfill
\vfill
\Large{Coucou}
\vfill
```

- Tracer une ligne horizontale traversant complètement la page : `\hrule`
- Ajouter des logos/images : suivre les notes du TP suivant où les moindres secrets des images vous seront dévoilés !
- `\vspace` permet de décaler la texte soit vers le haut (via un argument négatif) soit vers le bas (via un argument positif). Exemple : `\vspace{2cm}`.
- `\hline` offre la possibilité de tracer des lignes horizontales entre des morceaux de texte.

Ajouter une en-tête et un pied de page personnalisés

Le meilleur moyen de gérer les hauts/pieds de page est de faire appel au package **fancyhdr**. C'est un package très utilisé et très vaste. Voici un exemple de code à ajouter dans le préambule.

Utilisation de fancyhdr

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
% Première façon de passer les arguments
\fancyhead[LE,RO]{Overleaf}
\fancyhead[RE,LO]{Guides and tutorials}
\fancyfoot[CE,CO]{\leftmark}
\fancyfoot[LE,RO]{\thepage}

% Seconde façon de passer les arguments

\lhead{\bf{La belle en-tête}}
\chead{}
\rhead{\small{\textit{\leftmark}}}}
%\leftmark fait référence au plus grand numéro de section

\lfoot{}
\cfoot{}
\rfoot{\thepage / \pageref{LastPage}}
%faire référence à la dernière page demande le package "LastPage"
```

À l'inverse, si vous souhaitez qu'une page de votre document ne contienne pas les en-têtes et pieds de page, vous pouvez utiliser la commande `\thispagestyle{empty}`.

Pour ajouter la date à un endroit, on peut utiliser le package `datetime`, qui contient les commandes `\today` et `\currenttime`.

Sources :

https://fr.overleaf.com/learn/latex/Headers_and_footers#Standard_page_styles

<https://tex.stackexchange.com/a/91357>